JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 18, No. 5, September–October 1995

Aircraft Antilock Brake System with Neural Networks and Fuzzy Logic

H. Chris Tseng*
Duke University, Durham, North Carolina 27708-0291
and
Charlie W. Chi†
Santa Clara University, Santa Clara, California 95035

We explore the competitive concepts of neural network identification and fuzzy logic adaptive inference into the aircraft antilock brake system (ABS) design. The nonlinearities typically characterized by look-up tables are emulated with neural identifiers and subsequently integrated in the ABS design. The highly nonlinear aircraft dynamics coupled with varying operating conditions and uncertainties are handled well with the proposed rule-based fuzzy logic controller. The proposed methodology is shown to be robust under various runway ground conditions. The proposed aircraft ABS design with neural networks and fuzzy logic can persistently provide the maximum achievable traction force for deceleration purposes, as compared with other ABS and non-ABS schemes.

Nomenclature

D_A	= aerodynamic drag on aircraft, lb
F_D	= drag load force, lb
F_V	= vertical load on tire, 25,000 lb
K_D	= aerodynamic drag coefficient, 0.226 lb/(ft/s)^2
M	= total aircraft mass, slugs
m	= runway downhill slope, 0.0038
NB	= negative big
NM	= negative medium
NS	= negative small
PB	= positive big
PM	= positive medium
PS	= positive small
R	= set of real numbers
R^+	= set of positive real numbers
R^n	= nth-order vector with n real-number components
R_0	= radius of free tire, 1.85 ft
R_R	= rolling radius, ft
T_B	= brake torque, ft-lb
T_J	= engine thrust
V_{ac}	= velocity of aircraft, ft/s
V_S	= slipping velocity
V_W	= wind velocity
W	= total aircraft weight, 100,000 lb
x'	= rate of change of x with respect to time t
ZE	= zero
$\Delta x(k), k \in R^+$	= incremental change of x at instant k
δ_0	= tire deflection
μ_{max}	= maximum friction coefficient
$\mu_{max_{max}}$	= maximum pavement friction coefficient
μ_{rat}	= coefficient of friction ratio
ω	= wheel speed, rad/s

I. Introduction

ANTILOCK braking systems (ABS) have been widely used in both automobiles and aircraft. Due to the uncertain nature of the road conditions and the complicated nonlinear dynamics of the wheels coupled with the aircraft system, the ABS design imposes a challenging issue for control and aerospace engineers. Recently, a study revealed that the existing ABS designs in more than 10 million cars in the United States did not deliver what they promised and called for more research work.[1] Numerous companies including Boeing, McDonnell Douglas, and Ford Motor are actively investigating this subject.

The ever-increasing success of identification and classification of nonlinear systems with neural-network-related applications are reported in numerous control papers.[2–4] The generalization capability of neural learning is ideal for interpreting the nonlinear characteristics from limited data. Fuzzy logic has succeeded in many control problems where traditional approaches have failed because fuzzy logic comes to grips with the pervasive imprecision of the real world and offers a methodology for exploiting the tolerance for imprecision to achieve tractability, improved performance, and lower solution cost. The fuzzy-logic-based self-tuning proportional-integral (PI) controller by Tseng and Hwang[5] is one of the designs that demonstrates its superior performance over the conventional controller for a nonlinear system with modeled uncertainties and noises.

In this philosophy, we integrate fuzzy logic and neural networks with the ABS design and present our work as follows. In Sec. II, we present the problem statement. Fundamental concepts of neural networks and fuzzy logic are introduced in Sec. III, followed by their nonlinear interpolation of ABS data in Sec. IV. The rule-based fuzzy logic controller and its successful traction experiments for ABS are described in Sec. V. We end this paper in the conclusion section by providing direction for future research in aerospace control, suggesting areas where integration of modern control theory with artificial intelligence techniques can provide a powerful tool for the development of future intelligent controllers.

II. Problem Formulation

From the braking point of view, the aircraft or automotive vehicles need maximum traction force through the friction between the ground and the wheel to decrease the velocity. As the brake is being applied, the tangential speed of the wheel is lower than the speed of the aircraft and the slipping phenomenon develops. Slipping velocity generates the required traction force to achieve aircraft deceleration. Generally speaking, the relationship between the slipping and the induced friction is nonlinear and nonmonotonic. Maximum frictional traction is available when slipping velocity is around 10–20 ft/s depending on the ground conditions. As excessive braking torque is applied, frictional traction is reduced due to too high of a slipping velocity. This is commonly known as wheel lock-up. The problem is further complicated by the complex nonlinear dynamics of the aircraft coupled with the wheel system. It is

Received Nov. 10, 1994; revision received April 7, 1995; accepted for publication April 7, 1995.

*Associate Research Professor, Department of Electrical Engineering.
†Graduate Student, Department of Electrical Engineering, Intelligent Control Laboratory.

common that parts of the nonlinear dynamics are available through limited look-up tables. Identifiers are needed to complete the modeling task. Since the ground conditions are typically varying and unavailable, the ABS design has to be robust with respect to such. We summarize the problem statement as follows: A brake control is to be designed to maintain the desired slipping velocity between the wheel and the aircraft so that persistent maximum traction force is available to stop the aircraft.

III. Fundamentals of Neural Networks and Fuzzy Logic

The ever-increasing need for autonomous decision making in complex systems calls for methodologies specialized in adaptation and high machine IQ. Fuzzy logic is a natural methodology well suited for problems that contain vague or imprecise information. For instance, the varying runway conditions, changing cargo loads and the associated tire deflection, and the delay of brake action coupled with inconsistent sensor readings render the ABS problem an ideal setting for fuzzy logic application. The identification of nonlinear characteristics is important in modeling and controller design in the field of aerospace engineering. For nonlinear systems with uncertainties, neural networks and the proper learning rules are ideal settings for control or pattern classification of such. The widespread use of these new intelligent systems technologies in various fields indicates that the concepts are gaining acceptance.

Neural Networks and Identification

Artificial neural networks or simply neural networks are layers of clusters of interconnected neurons. Each neuron receives scaled inputs from other neurons as governed by the interconnection weights and transmits the signal after a smooth nonlinear filtering of the combined inputs. Consider the simplest basic unit of neural network, a single-input, single-output, neuron, as depicted in Fig. 1.

This fundamental unit shows how the input $u \in R$ is scaled with weight w and combines with the bias $b \in R$ to form the intermediate output v. The neuron output $y \in R$ is obtained as v is filtered through the sigmoid function $g(v)$. The choice of the sigmoid function is not unique, but it is typically smooth and bounded, such as

$$y = \frac{2}{\pi} \tan^{-1} \frac{\pi \lambda}{2} v \tag{1}$$

or

$$y = \frac{1 - e^{-\lambda v}}{1 + e^{-\lambda v}}, \qquad \lambda \in R^+ \tag{2}$$

The input–output characteristics of the neuron is obviously dependent on the choice of w and the sigmoid function. The feedforward neural network is shown in Fig. 2 and its compact representation in Fig. 3. In general, it consists of layers of interconnected neurons with each neuron as described in Fig. 1.

Overall the neural network can be viewed as a nonlinear mapping $f: R^n \rightarrow R^m$ relating the input–output pair $(x, y), x \in R^n$ and $y \in R^m$. This neural mapping relationship is depicted in Fig. 4. The mapping relationship induced by the neural network is dependent on the interconnection weight w. Through appropriate learning or training the neural network can be adjusted to emulate the given characteristics through weight adaptation. This is the basic principle of neural identification. The capability of the neural network to approximate nonlinear functions is summarized in the following existence theorem.

Theorem 1. Kolmogorov's neural network map existence theorem. Any given continuous function $f: x \rightarrow y$ where $x \in [0, 1]^n \subset R^n, y \in R^m$, can be implemented exactly by a three-layer neural network with n neurons in the input layer, $2n + 1$ neurons in the hidden layer, and m neurons in the output layer. The proof of this theorem can be found in Sprecher.[6]

Remark 1. This existence theorem only provides the sufficient conditions. It is of interest to comment that in many cases the neural networks have a number of layers or neurons inconsistent with the listed conditions and yet provide an excellent emulation.

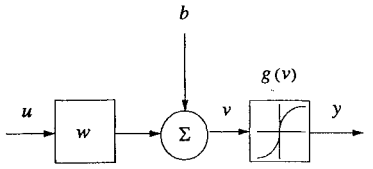

Fig. 1 Single-input single-output neuron.

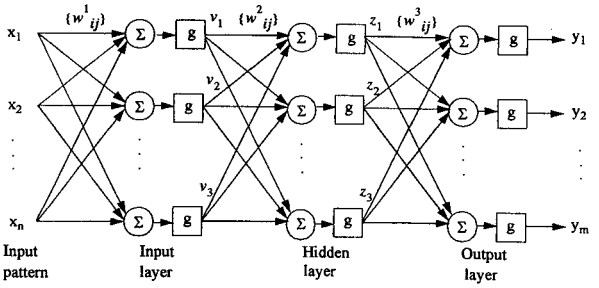

Fig. 2 Feedforward multilayer neural networks.

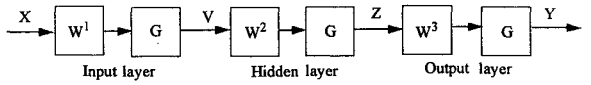

Fig. 3 Compact representation of neural networks.

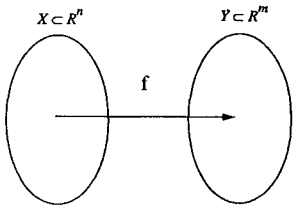

Fig. 4 Neural network as a nonlinear mapping.

Remark 2. There are numerous neural models besides the feedforward one.[7] For identification purposes, we shall concentrate on the feedforward model.

During training, the interconnection weights of the neural network are adaptively updated so as to follow the input–output data pair of the function f to be identified, $y = f(x), x \in R^n, y \in R^m$. Backpropagation learning, an efficient gradient method utilizing the cascaded layer structures of the neural network, will be used to train the neural network and can be represented as

$$W(k+1) = W(k) - \eta \frac{\partial J}{\partial W} \tag{3}$$

where $W \in R^P$ is the generalized interconnection weight vector, $\eta \in R^+$ is the learning rate, and J is the performance index

$$J(k) = \tfrac{1}{2} e^T(k) e(k) \tag{4}$$

and

$$e(k) = y^d - y(k) \tag{5}$$

where $y \in R^m$ is the neural network output and y^d is the output of the function to be identified. With extensive training, the resultant neural networks can emulate the given function and thus serve as an identifier. However, it is not unusual that the learning process may be terminated at local minima.

Fuzzy Logic in Control

Ever since Zadeh[8] introduced the concept of fuzzy set theory in 1965, it has found many applications in a variety of fields. One of the most important and successful applications of fuzzy set theory is the subject of fuzzy logic in control.[9]

Unlike a crisp set, fuzzy sets allows partial membership. For example, a sensor reading of 1.2 may have membership values of

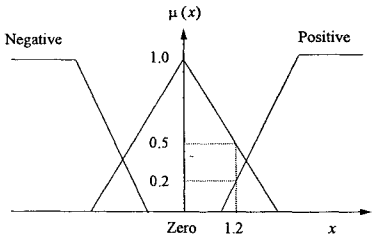

Fig. 5 Triangular fuzzy membership functions.

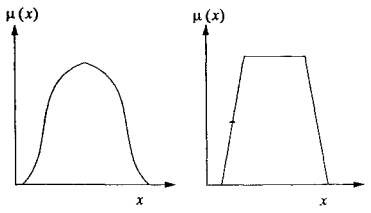

Fig. 6 Common membership functions.

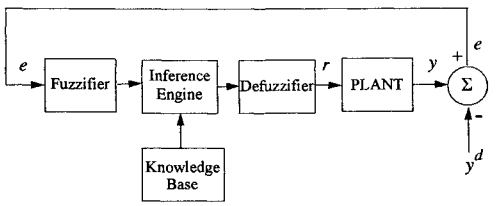

Fig. 7 General architecture of a fuzzy logic controller system.

0.5 and 0.2 for being zero and positive, respectively. The partial membership for the sensor reading x is characterized by the fuzzy membership functions shown in Fig. 5 and can be expressed as $\mu_{\text{positive}}(x) = 0.2$ and $\mu_{\text{zero}}(x) = 0.5$. Other popular membership functions may take one of the forms such as the Gaussian-type or trapezoidal as shown in Fig. 6. In our investigation, the Gaussian-type membership functions were used and have the form

$$\mu(x) = \exp\left[-\frac{1}{2}\left(\frac{x-\bar{x}}{\sigma}\right)^2\right] \tag{6}$$

where x is the input, $\bar{x}$ is the mean, and σ is the standard deviation of the membership function.

Figure 7 illustrates a basic architecture of the fuzzy logic controller. The numerical output data from the output sensor of the process under control is codified through the fuzzifier into the equivalent linguistic parameters such as {negative, zero, positive} with associated membership function values. Various rules in the knowledge base and the decision-making logic are invoked and contribute their control actions with different degrees of emphasis depending on its respective membership values. The inference engine is where the rules are evaluated. The fuzzy rules are derived from a typical servo response profile such as the one shown in Fig. 8. In particular, each rule is derived from the representative points of the response curve, such as the peaks and crossover points. For example, a typical rule for the control problem to achieve the set point tracking objective can be

If Error is NB and Change in Error is ZE

Then Change in Control is PB

where Error is defined as

$$e(t) = y^d - y(t) \tag{7}$$

and Change in Error is defined as

$$e'(t) = \frac{\mathrm{d}}{\mathrm{d}t}e(t) = -y'(t) \tag{8}$$

The 19 fuzzy rules shown in Table 1, which are described in detail by Lee[10] for fuzzy logic control, although limited, should be sufficient to satisfy the control objective. It is assumed that the control tends to suppress the output y. The sign of the Δu in Table 1 would have to be reversed if the control effort acts otherwise. The operating regions not specifically addressed by those 19 rules are covered through fuzzification.

Table 1 Fuzzy rule base

Rule no.	e	e'	Δu
1	NB	ZE	PB
2	NM	ZE	PM
3	NS	ZE	PS
4	ZE	PB	NB
5	ZE	PM	NM
6	ZE	PS	NS
7	PB	ZE	NB
8	PM	ZE	NM
9	PS	ZE	NS
10	ZE	NB	PB
11	ZE	NM	PM
12	ZE	NS	PS
13	ZE	ZE	ZE
14	NB	PS	PM
15	NS	PB	NM
16	PB	NS	NM
17	PS	NB	PM
18	NS	PS	ZE
19	PS	NS	ZE

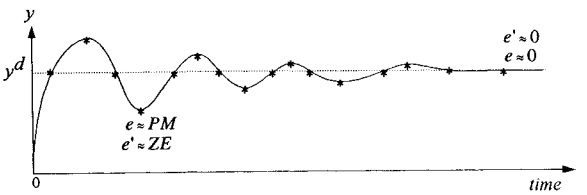

Fig. 8 Typical single output plant response for a set point at y^d.

The final stage in the fuzzy logic controller architecture aggregates all the inferred fuzzy control action using singletons for each consequent and produces a nonfuzzy numerical control through the defuzzification module. The centroid defuzzification is a common methodology and consists of functions of the form

$$\Delta u = \frac{\sum_{j=1}^{m} z_j \prod_{i=1}^{n} u_{A_j^i}(x_i)}{\sum_{j=1}^{m} \prod_{i=1}^{n} u_{A_j^i}(x_i)} \tag{9}$$

where $u_{A_j^i}(x_i)$ is the associated membership value of the ith input x_i for the jth heuristic rule, z_j is the equivalent numerical control action to be fired upon for the jth rule, and Δu is the resulting aggregated control change.

IV. Nonlinear Interpolation

Often only a limited number of data points is available from empirical measurements for certain runway characteristics, as shown in Tables 2 and 3. In the aircraft antiskid model, one has the tables of tire deflection δ_0 vs vertical load F_V and coefficient of friction μ_{rat} vs slipping velocity V_S. From the table entries, it is obvious that runway characteristics are both nonlinear and nonmonotonic.

Generalization with Neural Networks

Generalization can be obtained using a feedforward two-layer neural network structure similar to the one in Fig. 2. Basically, each table entry is viewed as an input–output data pair. The backpropagation learning algorithm was used to train the neural network. The number of training cycles n may depend on the initial choice of weights, as in our case. As the number of iterations increases, the emulation improves progressively. After extensive training, the neural network is able to pass through the given data pair or in a close neighborhood of the nonlinear interpolation curves, as shown in Figs. 9a and 9b. The nonlinear interpolation of Table 3 using neural networks is close to the experimental model by Zalovcik[11] shown in Fig. 9c and therefore is a good approximation of the associated nonlinear characteristics.

Table 2 Vertical load vs tire deflection

F_V (lbs)	δ_0 (ft)
0	0.0
10,000	0.104
30,000	0.213
110,000	0.667

Table 3 Coefficient of friction vs slipping velocity

V_S (ft/s)	μ_{rat}
0	0.0
13.43	1.0
25.54	0.94
60.31	0.667

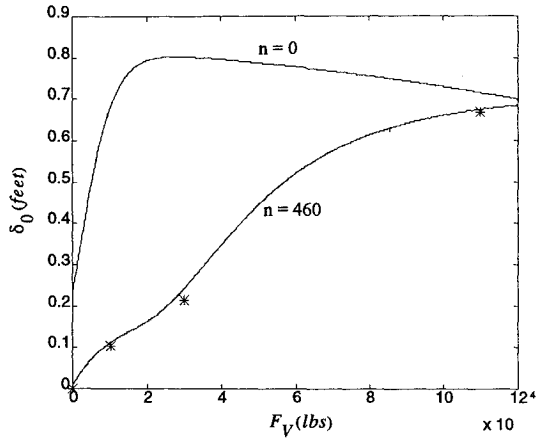

Fig. 9a Initial and final training cycles n of Table 2 using neural networks.

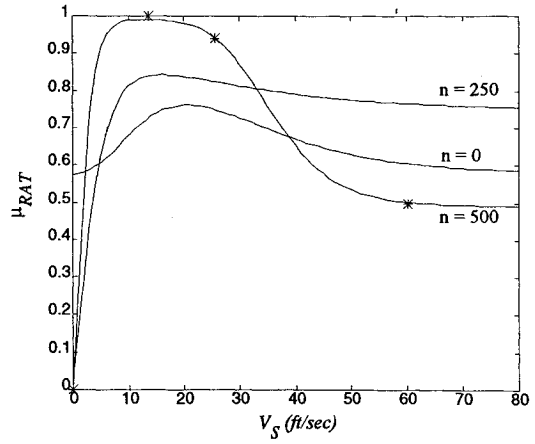

Fig. 9b Initial, intermediate, and final training cycles n of Table 3 using neural networks.

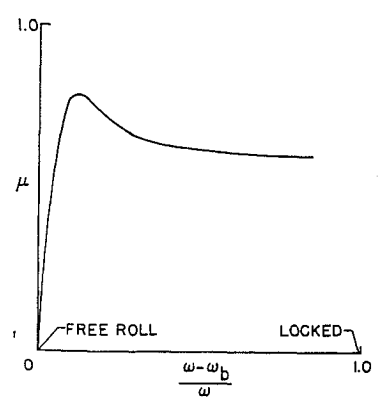

Fig. 9c Zalovcik's slipping velocity vs friction coefficient model under dry runway conditions.

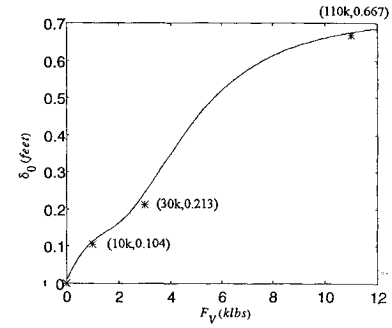

Fig. 10 Nonlinear interpolation of Table 2.

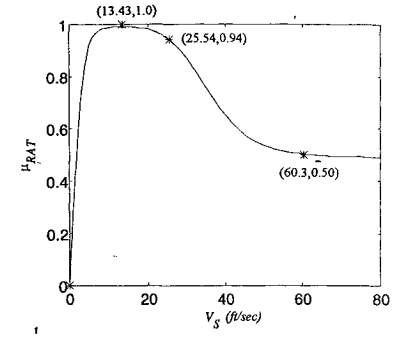

Fig. 11 Nonlinear interpolation of Table 3.

Generalization Using Fuzzy Logic

We can also use fuzzy logic to interpolate the nonlinearities. This technique is derived from the concept of a fuzzy graph.[12] Basically, each table entry is converted into a fuzzy rule. We then apply the fuzzification with the appropriate membership functions to the fuzzy rule base to obtain the nonlinear interpolation curves.

Since there are only four numerical data points available for Table 2, assign each tire deflection data a fuzzy membership denoted by ZE, PS, PM, and PB. The shape of each membership function is the Gaussian type described by Eq. (6) except the left and right spread are unsymmetrical. Table 2 is divided into four fuzzy membership regions where the mean of each membership function corresponds to the given vertical load for F_V.

The next task here is to convert the table entries into a set of fuzzy rules from the given data pairs. For each pair of data, we construct one corresponding if–then rule. Four such rules are constructed in this way:

1) **If** F_V is ZE **Then** δ_0 is ZE.
2) **If** F_V is PS **Then** δ_0 is PS.
3) **If** F_V is PM **Then** δ_0 is PM.
4) **If** F_V is PB **Then** δ_0 is PB.

The defuzzification of Eq. (9) is used to determine the tire deflection δ_0 due to a vertical load F_V as follows:

$$\delta_0 = \frac{\sum_{j=1}^{4} z_j \mu_j(F_V)}{\sum_{j=1}^{4} \mu_j(F_V)} \tag{10}$$

where $\mu_j(F_V)$ is the associated membership value of the jth heuristic rule and z_j is the equivalent numerical data associated with the jth table entry of the tire deflection.

Fuzzy logic has the generalization feature that is capable of providing the associated nonlinear characteristics beyond the look-up table entries, as shown in Fig. 10.

Similarly, a set of fuzzy if–then rules can be generated from the given data pairs in Table 3 as follows:

1) **If** V_S is ZE **Then** μ_{rat} is ZE.
2) **If** V_S is PS **Then** μ_{rat} is PB.
3) **If** V_S is PM **Then** μ_{rat} is PM.
4) **If** V_S is PB **Then** μ_{rat} is PS.

Here, V_S is the sensor measurement of the slipping velocity in feet per second. The Gaussian-type fuzzy membership is assigned to each linguistic description of the associating coefficient μ_{rat}.

Figure 11 shows the fuzzy-logic-based interpolation passes through a close neighborhood of each of the four entries from Table 3.

As a comparison, the neural network is able to generalize a smoother nonlinear curve than the fuzzy logic interpolation. However, a neural network requires extensive training that can be done off-line. On the other hand, fuzzy logic is much easier to implement, but it provides only a crude estimate of the nonlinear interpolation curves. For our modeling purposes, the neural network interpolation will be used in our ABS simulation.

V. ABS Design

We begin by presenting the aircraft dynamics coupled with the wheel system subject to braking as follows:

$$V'_{ac} = \frac{1}{M}[T_J - D_A(t) - F_D(t) + mW] \tag{11}$$

$$D_A(t) = K_D[V_{ac}(t) + V_W]^2 \tag{12}$$

$$F_D(t) = \mu_{rat}[V_S(t)]\mu_{max}F_V \tag{13}$$

$$\mu_{max} = \mu_{max_{max}} f[V_{ac}(t)] \tag{14}$$

$$\omega' = \frac{1}{I}\{F_D[\omega(t)]R_0 - T_B\} \tag{15}$$

$$V_S(t) = V_{ac}(t) - R_R\omega(t) \tag{16}$$

$$R_R = R_0 - \tfrac{1}{3}(\delta_0(F_V)) \tag{17}$$

In our investigation, the simplified ABS prototype design will focus on one main landing gear wheel. The numerical values used to simulate the aircraft dynamics are quoted from a set of DC-9 flight test data. From the simplified block diagram of the ABS system shown in Fig. 12, it is easy to see that this nonlinear system consists of interconnected subsystems, the aircraft subsystem and the wheel subsystem. In the aircraft dynamics, the neural network identifier is substituted for the look-up table of coefficient of friction vs slipping velocity. The maximum friction of coefficient is a function of airplane velocity for the various runway surface, and it can be obtained from a linear interpolated look-up table. For the wheel dynamics, another neural identifier is used to replace the look-up table of tire deflection vs vertical load. The fuzzy logic controller maintains the desired slipping velocity by issuing the proper brake torque control action. The brake control acts on the wheel dynamics and subsequently the aircraft speed through the drag force F_D.

The objective of this simplified ABS control problem is to maintain the highest possible μ_{rat} that occurs when the desired slipping speed $V_S^d = 13.43$ ft/s under dry runway conditions, as indicated in Table 3. Using Eqs. (13) and (17), we can calculate the corresponding desired wheel speed $\omega^d(t)$ that achieves maximum braking drag force subject to varying aircraft velocity as

$$\omega^d(t) = \frac{V_{ac}(t) - V_S^d}{R_R} \tag{18}$$

We define the difference between the wheel speed $\omega(t)$ and that of the desired speed $\omega^d(t)$ as

$$e(t) = \omega^d(t) - \omega(t) \tag{19}$$

The rate of change of e is given by

$$e'(t) = -\omega'(t) = (-1/I)\{F_D[\omega(t)]R_0 - T_B(t)\} \tag{20}$$

The equivalent control objective is to have the wheel speed $\omega(t)$ track a time-varying profile $\omega^d(t)$ that varies with the aircraft speed $V_{ac}(t)$. The input control T_B is being updated as

$$T_B(t+1) = T_B(t) + \Delta T_B(t), \qquad t = 0, 1, 2, \ldots \tag{21}$$

The fuzzy rule set as described in Table 1 will be used for updating ΔT_B iteratively.

For this particular problem, we used Gaussian-type fuzzy membership functions (6) for e and e', but ΔT_B takes singleton values. Table 4 summarizes the membership function description for e, e', and ΔT_B.

We now verify the fuzzy neural design by applying the proposed fuzzy logic controller above to the aircraft dynamics (11–16). The neural network identifiers of Tables 2 and 3 will be used to describe $\mu_{rat}[V_S(t)]$ and $\delta_0(F_V)$ in Eqs. (13) and (17), respectively. We assume the following initial state $\omega(0) = 109$ rad/s, $T_B(0) = 0$ ft-lb, and $V_{ac}(0) = 203$ ft/s.

Figure 13a indicates our ABS approach induces slip velocity in a close neighborhood of the desired one with a relatively small offset as compared to Stubbs et al.'s,[13] which exhibited significant offset around the threshold slip velocity, as shown in Fig. 13b.

The aircraft velocity profiles in Fig. 14 show the comparison between ABS with neural network and fuzzy logic and non-ABS under dry runway conditions. As we can see, the proposed ABS design was able to decelerate the aircraft to a reasonable speed in under 25 s as opposed to the non-ABS. The non-ABS approach

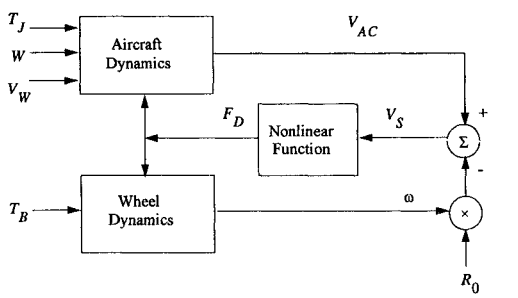

Fig. 12 Block diagram of interconnected aircraft and wheel dynamics.

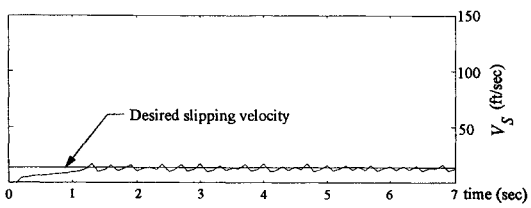

Fig. 13a Proposed ABS slipping velocity profiles.

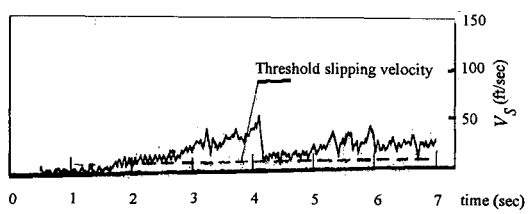

Fig. 13b Stubbs and Tanner's slipping velocity profiles.

Table 4 Fuzzy parameters description

	Range	σ	x(NB)	x(NM)	x(NS)	x(ZE)	x(PS)	x(PM)	x(PM)
e	−5, 5	0.8, 1	−5	−3.5	−1.5	0	1.5	3.5	5
e'	−100, 100	15, 20	−100	−65	−30	0	30	65	100
ΔT_B	−6000, 6000	n/a	−6000	−3750	−1750	0	1750	3750	6000

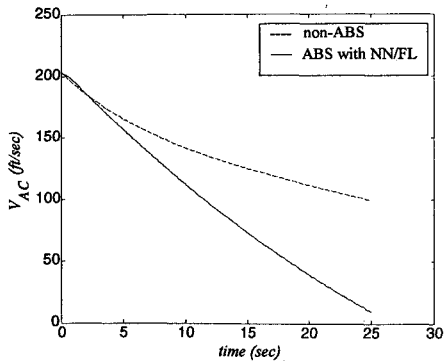

Fig. 14 Velocity profiles of proposed ABS and non-ABS design.

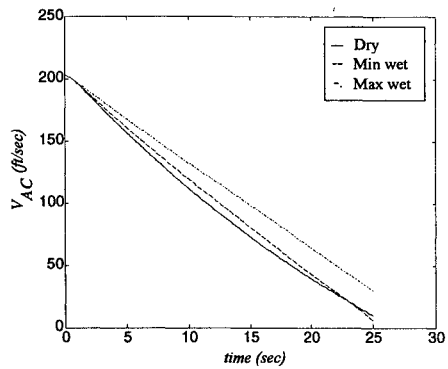

Fig. 15 Velocity profiles under various runway surface conditions.

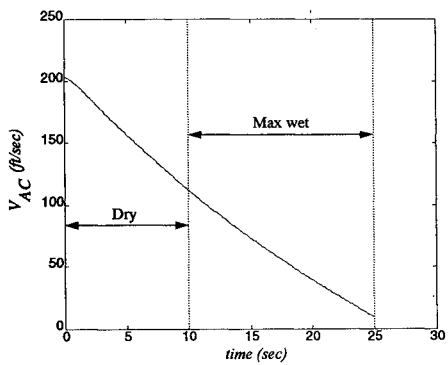

Fig. 16 Patchy runway conditions from dry to max wet.

utilizes a linear interpolated look-up table for brake torque as a function of aircraft velocity.

To illustrate the robustness of our proposed fuzzy logic controller, Fig. 15 shows the velocity profiles subject to various runway surface conditions. In all three cases with different runway surfaces, the aircraft decelerated to a very low speed in just under 25 s. The effects of "patchy" runway surface conditions from dry to maximum wet, as shown in Fig. 16, were also investigated. In the first 10 s, the aircraft experienced dry runway conditions followed by maximum wet. The result was almost equally satisfactory. The aircraft decelerated to a reasonably low speed in 25 s.

VI. Conclusion

Fuzzy logic and neural networks are shown to be successful tools in our simplified ABS problem. The uncertain and nonlinear features of the ABS find fuzzy logic as an ideal setting to cope with such. Other aerospace and vehicle control issues like the adaptive suspension systems, autonomous navigation, and engine control, just to name a few, will also find the proposed approach useful. Future research calls for the ABS design for the realistic system where the aircraft dynamics are coupled with the dynamics of several wheels. Blending the on-line learning ability of neural networks with the adaptive fuzzy inference to accommodate the delay in brake action is another important issue that requires more study.

Acknowledgments

The authors were supported by the National Science Foundation under Grant ECS-9110886. The second author thanks Douglas Dahlby for his valuable discussion and suggestions on this paper.

References

[1]The Highway Loss Data Institute, "Anti-Lock Brakes Don't Work as Well as Thought," *San Jose Mercury News*, Jan. 27, 1994.

[2]Narendra, K. S., and Mukhopadhyay, S., "Intelligent Control Using Neural Networks," *IEEE Control System Magazine*, Vol. 12, No. 2, 1992, pp. 11–18.

[3]Sutton, R. S., Barto, A. G., and Williams, R. J., "Reinforcement Learning in Direct Adaptive Optimal Control," *IEEE Control System Magazine*, Vol. 12, No. 2, 1992, pp. 19–22.

[4]Nguyen, D., and Widrow, B., "Neural Network for Self-Learning Control Systems," *IEEE Control System Magazine*, Vol. 10, No. 3, 1990, pp. 18–23.

[5]Tseng, H. C., and Hwang, V. H., "Servocontroller Tuning with Fuzzy Logic," *IEEE Transactions on Control Systems and Technology*, Vol. 1, No. 4, 1993, pp. 262–269.

[6]Sprecher, D. A., "On the Structure of Continuous Functions of Several Variables, *Transactions of the American Mathematical Society*, Vol. 115, March 1965, pp. 340–355.

[7]Hetch-Nielsen, R., *Neurocomputing*, Addison–Wesley, Reading, MA, 1990, pp. 79–109.

[8]Zadeh, L. A., "Fuzzy Sets," *Information Control*, Vol. 8, 1965, pp. 338–353.

[9]Mamdani, E. H., and Assilian, S., "An experiment in Linguistic Synthesis with a Fuzzy Logic Controller," *International Journal of Man-Machine Studies*, Vol. 7, No. 1, 1975, pp. 751–769.

[10]Lee, C. C., "Fuzzy Logic in Control Systems: Fuzzy Logic Controller–Parts I, II," *IEEE Transactions Systems, Man, Cybernetics*, Vol. 20, March/April 1990, pp. 404–439.

[11]Zalovcik, J. A., "Ground Deceleration and Stopping of Large Aircraft," AGARD Rept. 231, Oct. 1958, p. 18.

[12]Zadeh, L. A., "Similarity Relations and Fuzzy Ordering," *Information Sciences*, Vol. 3, 1971, pp. 177–200.

[13]Stubbs, S. M., and Tanner, J. A., "Behavior of Aircraft Antiskid Braking Systems on Dry and Wet Runway Surfaces: Slip-Velocity Controlled," *NASA Ames Journal*, 1979, p. 39.